



Obtenir et envoyer des données

Afin d'envoyer et de recevoir des données, Angular fournit un module HTTP.

Le module **HttpClient** expose plusieurs méthodes permettant d'envoyer et recevoir des données via des requêtes HTTP.

Éditer le fichier **app.module.ts** puis ajouter le module **HttpClientModule**

NB :

1- Il faut l'importer

2- Il faut l'ajouter dans la propriété imports de @NgModule

```
...
import {HttpClientModule} from '@angular/common/http'
...
@NgModule({
  ...,
  imports: [
    ...,
    HttpClientModule
  ],
  ..
})
export class AppModule { }
```

Créez un répertoire **models** dans **src/app**

Créez un fichier **post.model.ts** dans **src/app/models**

```
1 export class Post {
2   constructor(
3     public userId: number,
4     public id: number,
5     public title: string,
6     public body: string) {
7   }
8 }
```

Créez un répertoire **service** dans **src/app**

Créez un fichier **task.service.ts** dans **src/app/service**

```
1 import { Injectable } from '@angular/core';
2 import 'rxjs/add/operator/map';
3 import { HttpClient, HttpParams } from '@angular/common/http';
4 import { Observable } from 'rxjs/Observable';
5 import { Post } from '../models/post.model';
6
7 @Injectable()
8 export class TaskService {
9
10   baseUrl: string = "https://jsonplaceholder.typicode.com/posts"
11
12   constructor(private http: HttpClient) { }
13
14
15   getAllPosts(): Observable<Post[]> {
16     return this.http.get<Post[]>(this.baseUrl);
17   }
18
19   getPostById(id: string): Observable<Post> {
20     let params = new HttpParams().set('id', id);
21     return this.http.get<Post>(this.baseUrl, { params });
22   }
23 }
```

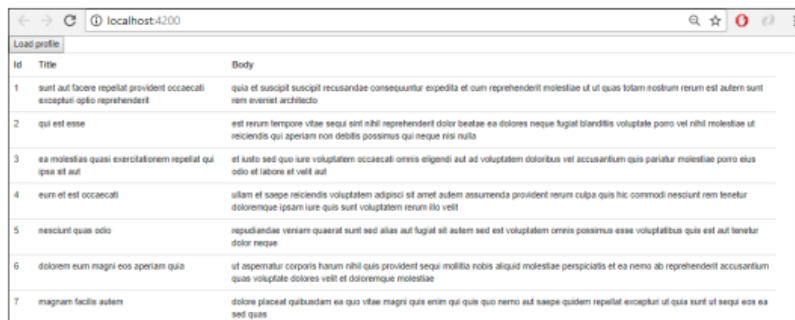
Éditer le fichier **app.component.ts** puis transformer

```
1 import { Component } from '@angular/core';
2 import { TaskService } from '../service/task.service';
3 import { Post } from '../models/post.model';
4
5 @Component({
6   selector: 'app-root',
7   templateUrl: './app.component.html',
8   styleUrls: ['./app.component.css']
9 })
10 export class AppComponent {
11
12   constructor(private taskService: TaskService) {
13   }
14
15   posts: Post[] = [];
16
17   //Fetch all articles
18   getAllPosts() {
19     this.taskService.getAllPosts()
20       .subscribe(
21         data => this.posts = data,
22         errorCode => console.log(errorCode)
23       );
24   }
25 }
```

Éditer le fichier **app.component.html** puis transformer

```
1 <div>
2   <button (click)="getAllPosts()">Load profile</button>
3 </div>
4 <table class="table">
5   <tr><th> Id</th> <th>Title</th><th>Body</th><th></th><th></th></tr>
6   <tr *ngFor="let post of posts" >
7     <td>{{post.id}}</td> <td>{{post.title}}</td> <td>{{post.body}}</td>
8   </tr>
9 </table>
```

Nous obtenons



The screenshot shows a web browser window with the address bar displaying 'localhost:4200'. The page title is 'Load profile'. Below the title, there is a table with 7 rows and 3 columns: 'Id', 'Title', and 'Body'. The table contains placeholder text for each row.

Id	Title	Body
1	sunt aut facere repellat provident occaecati excepteur optio reprehenderit	quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto
2	qui est esse	est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque ris nulla
3	ea molestias quasi exercitationem repellat qui ipsa est aut	et kusto sed qui iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium quis pariatur molestiae porro eius odio et labore et velit aut
4	eum et est occaecati	ullam et saepe reiciendis voluptatem adipisci sit amet autem assumenda provident rerum culpa quis hic commodi nesciunt rem tenetur doloremque ipsam iure qui sunt voluptatem rerum illo velit
5	nesciunt quae odio	repudiandae veritatis quaeerat sunt sed alias aut fugiat sit autem sed est voluptatem omnis possimus esse voluptatibus quis est aut tenetur dolor neque
6	dolorem eum magni eos aperiam quia	ut aspernatur corporis harum nihil quis provident sequi mollitia nobis aliquid molestiae perspiciatis et ea nemo ab reprehenderit accusantium quas voluptate dolores velit et doloremque molestiae
7	magnam facis audem	dolore placeat quibusdam ea quo vitae magni quis enim qui quis quo nemo aut saepe quidem repellat excepteur ut quia sunt ut sequi eos ea sed quas